Northwestern University

McCormick School of Engineering
Department of Electrical and Computer Engineering

CE 347 - Microprocessor System Projects
Spring 2023

Descriptive AI Glasses v1.0.0
Product Manual

Will Phillips, williamphillips2023@u.northwestern.edu

George Mousa, georgemousa2024@u.northwestern.edu

Haolan Zhan, haolanzhan2023@u.northwestern.edu

Table of Contents

Introduction

Our mission is to empower individuals with visual disabilities or blindness to be more aware of the world around them through innovative AI technology. Our device will accurately and quickly describe the scene in front of the user via audio, enabling them to navigate and understand their environment with confidence. Our goal is to provide a reliable and accurate scene description in less than one second. With this device, we aim to enhance the independence and quality of life for those who are visually impaired by reducing frustration and worry. Our target users are people with limited vision or permanent blindness who would benefit from an audible description of their environment.

Several apps are being developed to address this problem, including offerings from Microsoft, Google, and more. These apps may be perceived as intrusive, however, as they require the user to hold their phone in front of them constantly when describing the scene. Our device will instead reduce the cognitive burden on the user by conveniently and consistently performing its task through an easy-to-use and intuitive design that does not make it immediately obvious that the user is benefitting from a scene description device. We will create a set of descriptive AI glasses that can do this task, combining an advanced processor, camera, and AI system.

This document provides an overview of how to build and use the first version of the product. Granted, this version is very much a prototype. However, this version is able to demonstrate a proof-of-concept of using glasses to take a picture, transfer the image wirelessly, run the image through an A.I. image captioning model, and finally output an audio description to the user. Future iterations of the prototype will be able to miniaturize the captioning model to be able to run on a mobile application instead of a laptop, transfer the image wirelessly with greater speed, have an audio system built into the glasses itself, and have increased security in the process of pairing to the device with a mobile application.

Hardware Requirements

1. 1x Nicla Vision board
2. 1x Small Protoboard or Perfboard
3. 1x Push Button
4. 1x 330 Ohm Resistor
5. 1x LiPo battery with corresponding connectors
6. 3x Colored Wire (Only one color is necessary, but red (power), black (gnd), and some other color (push button) are preferred for organization)
7. 1x Glasses Frames
8. 1x Laptop capable of running both the AI captioning model as well as the BLE connection software
9. 1x Bluetooth Headset
10. 1x Double Sided Tape

Software Environment Setup/Maintenance

11. Download Arduino IDE 2 from https://www.arduino.cc/en/software. This is required to flash firmware onto the Nicla Vision.
    a. This step is not necessary if the complete prototype was purchased, since the firmware is already installed. But if updates or customization is desired, Arduino IDE 2 is required.
12. If building the prototype from scratch, follow this tutorial (https://docs.arduino.cc/software/ide-v1/tutorials/getting-started/cores/arduino-mbed_nicla) to download the Mbed OS Nicla core, which requires Arduino IDE 2.
    a. Skip this step if the complete prototype was purchased.
13. If building the prototype from scratch, update the bootloader: go to "File > Examples > STM32H747_System > STM32H747_manageBootloader" and upload this sketch to your board. After the sketch is uploaded, follow the instructions in the Serial Monitor.
    a. Skip this step if the complete prototype was purchased.
14. If building the prototype from scratch, clone or download the following arduino sketch from this GitHub link, and flash the Nicla Vision with the firmware.
    a. https://github.com/haolanzhan/ce347_glasses/blob/main/src/ble_peripheral_camera/ble_peripheral_camera.ino
    b. Skip this step if the complete prototype was purchased.
    c. Edit this file to update or improve on the firmware for the product.
15. If the above steps are completed, the Nicla Vision should have the required firmware to advertise itself as a BLE peripheral device. As long as power is delivered to the Nicla Vision, the firmware should be running, and awaiting a pairing with a BLE central device.
16. Download the following BLE Central python script to run on a local computer.
    a. https://github.com/haolanzhan/ce347_glasses/blob/main/src/ble_central_notify_image.py
    b. Requires the following Python Libraries (should be able to pip install):
        i. asyncio
        ii. io
        iii. bleak
        iv. pil
17. The software should be ready to run.

Imaging Captioning Setup/Maintenance

1. Download the pretrained image captioning model from the following link:
    a. https://drive.google.com/drive/folders/1ikeME8ktv_2v7cGVgFLOtgouWDxQMany
    b. Google Drive will break the folder into 12 zip files to download, due to the large folder size. Google Drive will append a number 1-12 to each zip file.
    c. Unzip the first zip file to use as the main folder.
    d. For each zip file 2-12, it will unzip into a folder containing files missing from the main folder. For each path in each of the aforementioned zip files, move all the individual files into the same path in the main folder (should only be image files in the two folders in the datasets/COCO/images path).
2. Connect to Local Host on Google Colab (helpful tutorial linked)
    a. https://research.google.com/colaboratory/local-runtimes.html
3. Once you have connected to a local runtime on Google Colab, cd into the directory where you want to set up the image captioning (same directory as the folder downloaded in step 1).
4. If it is your first time running the code, run all the blocks in Google Colab, including the ones that say “Run first time only!”
    a. Once you get to the blocks that use the gdown library, you might have to manually download the files from Google Drive, and place them in the folder you are working in.
        i. For the block that starts with print(reward), download the file from the URL that corresponds to the reward that is printed, and place it in the save/clipRN50_clips_grammar/ folder.
        ii. For the next block, download the file (cocotalk.json) from https://drive.google.com/uc?id=1HNRE1MYO9wxmtMHLC8zURraoNFu157Dp and place it in the data folder.
5. If it is not your first time running the code, only run the blocks that do not start with “Run first time only”.
6. Make sure that all the blocks were executed properly before running the final block, which will run the image captioning code on an image placed into the main folder you are working in.
    a. You can change which folder it will look for new images in by changing the folder_path variable.
7. The image should be displayed in the output, along with the corresponding text description and audio output.

Instructions to Build

1. Lay out circuit on protoboard
    a. The GND (or -) connection of the LiPo should connect to GND of the pushbutton circuit (in this case, one leg of the pushbutton) as well as have an output wire to GND of the Nicla Vision board
    b. The Vcc (or +) connection of the LiPo should connect only to an output wire which will go to the VBAT pin on the Nicla Vision board
        i. If the LiPo connector has an NTC pin, add an extra output wire from it to the NTC pin on the Nicla Vision. Otherwise, ignore it.
    c. The VDDIO_EXT pin from the Nicla Vision should be wired into the Vcc connection for the pushbutton circuit (connected to one leg of the 330 Ohm resistor)
    d. The remaining leg of the resistor and the remaining leg of the pushbutton should be connected to an output wire which will go to any GPIO in pin on the Nicla Vision (in our code, it is GPIO_0, but any pin will suffice as long as the code accounts for it)
2. Once the circuit has been laid out, try to shrink the circuit into the minimum size possible on the protoboard. Make sure the push button is accessible, as well as the input port for the LiPo battery.
3. Solder the circuit together, leaving long lengths of output wire.
4. Once the circuit has been soldered, cut off any excess unused protoboard.
5. Figure out your ideal positioning for the protoboard, LiPo, and Nicla Vision board on the glasses frame. The USB port on the Nicla Vision board should point upward, and the camera should point forward. The protoboard works best on the side of the glasses. Both work best at a right angle to each other. Make sure your layout works with the glasses frame and will not get in the way of the headset when attached.
6. Cut, strip, and solder the wires coming from the protoboard to the Nicla Vision at the correct length for your layout.
7. Attach the circuit to the glasses frame using double sided tape (or any preferred mounting method). For the battery, if using the Nicla Vision as the only charger and if not swapping batteries, you may directly mount the LiPo to the frame, though it is recommended that some kind of enclosure be used to hold the battery so that it may be taken out (in our case, the cardboard box the battery was stored in worked well)

Instructions to Use

1. Run each executable code on the following Google Colab shared file on the central device (laptop):
    a. https://colab.research.google.com/drive/1A0ORdbbCyaS6m-L68gvK1Y_bCSlu7dVa?usp=sharing
    b. The last code block in the above link should start the script that checks for new images that the central device receives via BLE, run it through the captioning model, and finally outputs audio.
2. Make sure the Nicla Vision/Glasses are connected to the battery
3. Run the python central script on your laptop:
    a. https://github.com/haolanzhan/ce347_glasses/blob/main/src/ble_central_notify_image.py
4. What to look for:
    a. When the Nicla vision is receiving power, but before it has been paired with a central device, it should blink 3 times repeatedly with a delay in-between.
    b. If the device is not paired, and the Nicla Vision is blinking continuously without delay, then there has been an initiation failure with either the BLE or camera hardware on the Nicla Vision, which will require the Arduino IDE to debug.
    c. Once the python central script is running, the computer should connect to the Nicla Vision. At this point, the Nicla Vision should blink continuously, signaling that the python script is currently attempting to subscribe to the Image Transfer BLE service that the Nicla Vision is advertising.
    d. Once the python script has subscribed to the Image Transfer Service, the Nicla Vision will stop blinking.
    e. Throughout this process, the central python script should be printing out statuses that can help with debugging.
5. Once the Nicla Vision is no longer blinking, the product is ready to use. Press a button on the glasses to take a picture. The LED blink once, signaling that the photo has been taken and is currently transferring.
6. After 7 seconds, the python script should have processed the image, and Google Collab will automatically run the image captioning model. Once processed, the computer will output the caption as audio.